Remarks:

Reconsideration of the application is respectfully requested.

Claims 1 - 26 are presently pending in the application. No claims have been amended or canceled. Claim 18 has been indicated as being allowable if rewritten to include all the limitations of the claims from which that claim depends.

In paragraph 4 of the above-identified Office Action, claims 1 and 6 - 13 were rejected under 35 U.S.C. § 103(a) as allegedly being obvious over Applicants Admitted Prior Art ("**AAPA**") in view of U. S. Patent No. 5,946,219 to Mason et al ("**MASON**"). In paragraph 11 of the Office Action, claims 2 - 5, 14 - 17 and 19 - 26 were rejected under 35 U.S.C. § 103(a) as allegedly being obvious over **AAPA** in view of **MASON** and further in view of U. S. Patent No. 6,077,315 to Greenbaum et al ("**GREENBAUM**").

Applicants respectfully traverse the above rejections.

Before discussing the prior art in detail, it is believed that a brief review of the invention as claimed, would be helpful. Claim 1 recites:

> "1. A method for configuring a configurable hardware block, the method which comprises:

(a) **implementing one of commands and command sequences of a program to be executed, the implementing step includes:**

(a1) **ascertaining** a given type of subunit of a configurable hardware block, the given type of subunit being required for executing a respective command;

(a2) **selecting**, if available, a subunit of the given type of subunit;

(a3) **configuring configurable connections** provided around the subunit selected in the selecting step, **if the subunit of the given type of subunit is found in the selecting step**;

(b) ascertaining configuration data with the step of implementing the one of commands and command sequences; and

(c) configuring the configurable hardware block by using the configuration data." [emphasis added by Applicants]

Page 29 of the instant application, line 10 - page 30, line 6, describes steps (a1) - (a3), as follows:

"In the first phase, **the type of virtual unit** (adder, subtracter, multiplier etc.) **required for executing the instruction in question is ascertained, and whether such a virtual unit is still available**. If a virtual unit of the required type is still free, this unit or one of these units is selected for executing the instruction in question. **Configuration** or preparation therefor **is then carried out** and the physical subunit associated with the selected virtual unit is reserved. For the purposes of configuration, the configuration bits associated with the physical subunit in question are simply set or reset. This poses no problems, because the information regarding which physical subunit has the selected virtual unit associated with it, via which configuration bits and how this physical subunit is to be configured, if appropriate, is, of course, managed together with the virtual unit. The physical subunit associated with the selected virtual unit needs to be reserved in order to prevent it from

> being possible for the physical subunit in question to be used more than once. In the example under consideration, this is achieved by virtue of the fact that, whenever a physical subunit has been allocated for a particular purpose, all the virtual units associated with the physical subunit in question are disabled." [emphasis added by Applicants]

The "first phase" or **implementing** step of Applicants' claim 1 is additionally followed by a "second phase", which is set forth in steps (b) and (c) of claim 1. Applicants' "second phase" is described in the instant application, page 30, line 19 - page 31, line 13, which states:

> **"In the second phase of hardware block configuration, the multiplexers connected upstream and/or downstream of the selected physical subunits are configured in order to set the data and/or signal sources and the data and/or signal destinations as per the stipulations in the instructions which are to be implemented.** The multiplexers and the format of the instructions which are to be implemented are, ideally, matched to one another such that those parts of the instructions which stipulate the data and/or signal sources, and those which stipulate the data and/or signal destinations, can be adopted unchanged as the configuration bits which configure the multiplexers. If--for whatever reason--this is not possible, the configuration bits configuring the multiplexers can be taken from a table, for example, which stores the association between the parts of the instructions which stipulate the data and/or signal sources and the data and/or signal destinations and the configuration bits which configure the multiplexers. The configuration which is required in order to produce a connection to a particular data and/or signal source and/or to a particular data and/or signal destination is preferably the same for all multiplexers." [emphasis added by Applicants]

As such, Applicants' claim 1 additionally includes, among other limitations, in steps (b) and (c), among other limitations, a **second ascertaining and configuring steps.**

The **AAPA**, neither teaches, nor suggests, among other limitations, Applicants' particularly claimed **implementing step**, including the particularly claimed **ascertaining, selecting and configuring** steps (a1), (a2) and (a3) of Applicants' independent claim 1. This is admitted in the Office Action, on page 3, which states:

> "The **AAPA does not explicitly teach**:
>
> e. **ascertaining a given type of subunit** of a configurable hardware block, the given type of subunit being required for executing a respective command
>
> f. **selecting, if available, a subunit** of the given type of subunit
>
> g. **configuring connections** provided around the subunit selected in the selecting step **if the subunit of the given type of subunit is found in the selecting step**
>
> In summary, **AAPA** does not explicitly teach configuring a selected portion of a subunit (if available) located with the configurable hardware block." [emphasis added by Applicants']

As the Office Action admits, the **AAPA** doesn't teach all of the claimed elements of Applicants' invention. However, as will be shown herebelow, the cited **MASON** and **GREENBAUM** references

additionally fail to teach or suggest the above limitations, among others, of Applicants' claim 1.

More particularly, the Office Action alleges, on page 3, that:

> Mason explicitly teaches a configuring a selected portion of a subunit [col. 1 line 45 - col. 2 line 41]. Furthermore, it is obvious that the portion to be configured would only be selected if it were available in order to prevent the subunit from malfunctioning. It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the device taught by Mason into the AAPA system because as Mason explicitly states, it would allow the device to be reconfigured "without having to program the entire device" which obviously saves an [sic] time [col. 1 line 67 - col. 2 line 1].

Applicants' respectfully disagree with what is allegedly taught in **MASON**. **MASON** does not teach or suggest, Applicants' particularly claimed **implementing** step including **ascertaining, selecting** and **configuring connections based on the selection.** This is made clear from a reading of the portion of **MASON** that is cited in the Office Action. Col. 1, line 45 - col. 2, line 41 of **MASON** states:

> "Continued evolution of programmable logic devices has resulted in the development of re-programmable interconnects, and more importantly the development of configurable logic cells. As the name implies, a configurable logic cell permits a designer to program the cell to function as any one of a number of basic logic gates or higher level logic functions. Current manufacturing technology makes possible the production of high density devices having many thousands of configurable logic cells and their associated interconnects, known as field programmable gate arrays (FPGA's). The availability of these high density

devices enables the designer to employ increasingly complex logic functions. Like their predecessors, FPGA's include programmable interconnects. Moreover, the interconnects are reprogrammable, further increasing the utility of FPGA's.

Reconfiguring these reprogrammable FPGA's, however, has typically required that the entire device be reconfigured. A further evolutionary step of these devices is represented in an FPGA manufactured by Atmel Corp., assignee of the present invention. Known as dynamically reconfigurable FPGA's, these devices allow only selected portions of the logic array to be reconfigured. In this way changes can be made to an FPGA without having to program the entire device, permitting only the selected portions of the array to be reconfigured.

Referring to FIG. 1, a typical FPGA 100 includes a plurality of configurable logic cells 130, configurable I/O blocks 110, and configurable inter- connects 120, 122, collectively referred to as resources of the FPGA. Although the interconnects 120, 122 are shown as a grid of individual interconnect lines, each "line" in actuality is a set of interconnect lines, as shown in FIG. 3B for instance. Each logic cell 130 and I/O block 110 includes data lines 140, 142 which can be selectively coupled to the interconnects 120, 122.

A typical design cycle begins with the design of one or more logic circuits which will then be implemented in an FPGA. A logic design includes logic gates and interconnections among the logic gates. Certain designs, however, such as digital filters incorporate the use of "constants," namely strings of ones and zeroes, to define their behavior. For the purposes of the description of the present invention, such constants are regarded as being part of the design and also will be referred to as logic gates.

For example, FIG. 2 shows a simple logic design. Each element in the logic design is identified by an instance name. Thus, the AND gates and the OR gate in FIG. 2 are named, G1-G3. FIG. 3A shows how the logic design might appear in the FPGA 100'. **Each of the gates G1-G3 and the interconnections shown in FIG. 2 is mapped to selected logic cells and interconnects as shown in FIG. 3A. Similarly, the inputs A-D and the output OUT (FIG. 2) are mapped to selected I/O blocks. Thus, interconnects 120a-120c and 122a-122c (shown**

> **high-lighted) connect together logic cells G1-G3 and I/O blocks 110a-110e. A magnified portion of the configuration of FIG. 3A is shown in FIG. 3B, illustrating the specific interconnections among the various logic cells, interconnects, and I/O blocks.** Although the design in the figures do not show the use of constants, it is known that logic cells in a modern FPGA can be configured to output a logic "1" or a logic "0" and that groups of logic cells can be so configured to produce one or more strings of ones and zeroes as needed." [emphasis added by Applicants].

Although **MASON** discloses that:

> **"Each of the gates G1-G3 and the interconnections shown in FIG. 2 is mapped to selected logic cells and interconnects as shown in FIG. 3A."**

It neither teaches, nor suggests, Applicants' limitations of claim 1 reciting:

> "(a1) **ascertaining** a given type of subunit of a configurable hardware block, the given type of subunit being required for executing a respective command;
>
> (a2) **selecting**, if available, a subunit of the given type of subunit;
>
> (a3) **configuring configurable connections** provided around the subunit selected in the selecting step, **if the subunit of the given type of subunit is found in the selecting step**; " [emphasis added by Applicants]

In fact, the Office Action, in order to find these elements in **MASON**, must fall back on making **an unsupported assumption** about **MASON**. In this regard, the Office Action only states:

> "Furthermore, **it is obvious** that the portion to be configured would only be selected if it were available in order to prevent the subunit from malfunctioning."

However, making such an assumption does not correspond to Applicants' particularly claimed **implementing** step (a), including the **ascertaining, selecting and configuring connections** steps (a1) - (a3).

That **MASON** neither teaches, nor suggests, Applicants' particularly claimed **ascertaining** step (a1), was argued in the previous Office Action Response, that Response being incorporated herein in its entirety. More particularly, Applicants' pointed out that **ascertaining**, as used in the present claims "implies that 'the given type of subunit' **is discovered or determined through some experimentation or examination** based on the requirements 'for executing a respective command' relative to available subunits in the configurable hardware block."

In the Office Action, it is confirmed that the above meaning has not been given to Applicants' claimed **ascertaining step (a1)**. More particularly, on pages 11 - 12 of the Office Action, item 33, it is confirmed that:

> "In response to argument (8), as stated above in the response to argument (5) in which applicant's argue that the references fail to show certain features of applicant's invention, it is noted that the features

> upon which applicant relies (i.e., "ascertaining" implies that "the given type of subunit is discovered or determined through some experimentation or examination based on the requirements for executing a respective command relative to available subunits in the configurable hardware block") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). **Therefore, the term ascertaining with respect to "ascertaining a given type of subunit" has not been interpreted by the examiner as being "discovered or determined through some experimentation or examination." Rather the examiner had interpreted ascertaining as "to apprise" which is the same as informing or giving notice to.** It should be apparent that the AAPA-Mason system does indeed give notice to or inform the functional unit 42 as to how it should be configured through the configuration data." [emphasis added by Applicants]

Applicants' respectfully traverse the application of a definition of "ascertain" different from that indicated in the Office Action. Applicants' believe it is improper to, on the one hand, acknowledge that claim terms are read in light of the specification, and, on the other hand, fail to do so. Applicants' specification **plainly** defines and uses the term "ascertaining" of step (a1) as **"discovered or determined through some experimentation or examination"**. For example, page 6 of the instant application, line 22 - page 7, line 7, states:

> "In other words, the hardware block is configured using configuration data which result from implementation of commands or command sequences of a program which is to be executed, and implementation of the commands or command sequences involves the following steps being carried out:

> - the type of configurable hardware block subunit which is required for executing a respective command is **ascertained**,
>
> - **a subunit of the type ascertained beforehand** which is still not being used otherwise is selected and - **provided that it has been possible to find such a subunit** -,
>
> - configurable connections provided around the selected subunit are configured." [emphasis added by Applicants]

Because of the way in which the term **ascertaining** of step (a1) has been used in the specification, it is unnecessary to read limitations from the specification into the claims, as alleged in the Office Action. By using the term "**ascertaining**" in Applicants' claims, **that term necessarily being interpreted in light of the specification**, the interpretation of **ascertaining** as "**discovered or determined through some experimentation or examination**" is already present in Applicants' claims. In light of the foregoing, Applicants' respectfully request reconsideration of the Examiner's usage of the unrelated definition of "**to apprise**" in reviewing Applicants' claims.

Further, because the **MASON** reference does not "**ascertain**" a subunit, as required by Applicants' claimed invention, it necessarily cannot perform Applicants' particularly claimed **selecting a subunit** step of (a2). Using the Examiner's definition of "to apprise" in place of Applicants' **ascertaining** (determining) step, eliminates Applicants'

particularly recited **selecting step**, i.e., if the ascertaining step (a1) were "to apprise" the processor of the available subunit, there is no need to **select** a subunit, as the system was already "**apprised**" of the usable subunit. The argument made in the Office Action's is different from Applicants' claimed **ascertaining** (finding) of available subunits, and then **selecting** one from those **found**.

As such, Applicants' believe that the **MASON** reference, alone, or in combination with the **AAPA** and **GREENBAUM** references, fails to teach or suggest Applicants' particularly claimed invention of claim 1.

Further, Applicants' independent claim 24 is additionally patentable over the references cited in the Office Action. Applicants' independent claim 24 recites:

> "24. A method for configuring a configurable hardware block, the method which comprises:
>
> **attempting to form pseudo-hyperblocks** including a plurality of hyperblocks **when implementing commands as configuration data**; and
>
> **configuring a configurable hardware block** by using the configuration data."

In item 11 of the Office Action, Claims 2 - 5, 14 - 17 and 19 - 26 were rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over the **AAPA** and **MASON**, as applied to claims 1

and 6 - 13, and further in view of **GREENBAUM**. Item 21 of the Office Action, merely states:

> "Referring to claims 24 - 26, these are rejected on the same basis as set forth hereinabove."

Applicants' respectfully traverse the above rejection of claim 24.

The **"attempting"** step of Applicants' claim 24 is disclosed on page 72 of the instant application, as follows:

> "Implementation of commands as configuration data therefore involves **attempting to form pseudo-hyperblocks where possible in a first step**. **This requires examining the program structure to determine whether pseudo-hyperblocks can be formed**, and carrying out an if conversion for the program parts which can be used to form pseudo-hyperblocks." [emphasis added by Applicants]

Applicants' believe that none of the cited references teach or suggest the claimed "**attempting**" step, which require, **examining** the program structure **to determine whether** pseudo-hyperblocks can be formed. Again, Applicants' **attempting** step, must be read in light of the specification of the instant application.

In response to Applicants' claim 21, which also recited an "**attempting**" step (i.e., attempting to form pseudo-hyperblocks including a plurality of hyperblocks when implementing

commands as configuration data), the Office Action stated, in item 18:

> "Referring to claim 21, Mason teaches that not only can basic logic be realized but "higher level logic functions" can be realized as well [col. 1, lines 48 - 51]. Because during loop unrolling the instructions are broken into blocks, it would have been obvious to create pseudo-hyperblocks that include a plurality of hyperblocks."

The above rejection of a similar limitation in Applicants' claim 21, does not address the **possible contingency** inherent in Applicants' claims 21 or 24 brought in by Applicants' claimed **"attempting"** limitation. None of the references cited teaches or suggests **attempting** to form pseudo-hyperblocks, i.e., **examining** the program structure **to determine whether** pseudo-hyperblocks can be formed, where such attempts may be unsuccessful, as a result.

It is accordingly believed that none of the references, whether taken alone or in any combination, teach or suggest the features of Applicants' claims 1 and 24. Claims 1 and 24 are, therefore, believed to be patentable over the art. The dependent claims are believed to be patentable as well because they all are ultimately dependent on claims 1 or 24. As it is believed that the claims were patentable over the cited art in their original form, the claims have not been amended to overcome the references.

Finally, Applicants appreciatively acknowledge the Examiner's statement that claim 18 "would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims." In light of the above, Applicants respectfully believe that rewriting of claim 18 is unnecessary at this time.

In view of the foregoing, reconsideration and allowance of claims 1 - 26 are solicited.

In the event the Examiner should still find any of the claims to be unpatentable, counsel would appreciate receiving a telephone call so that, if possible, patentable language can be worked out. In the alternative, the entry of the amendment is requested, as it is believed to place the application in better condition for appeal, without requiring extension of the field of search.

If an extension of time for this paper is required, petition for extension is herewith made.

Please charge any fees that might be due with respect to Sections 1.16 and 1.17 to the Deposit Account of Lerner and Greenberg, P.A., No. 12-1099.

Respectfully submitted,

Kerry P. Sisselman
Reg. No. 37,237

For Applicants

KPS:cgm

April 11, 2005

Lerner and Greenberg, P.A.
Post Office Box 2480
Hollywood, FL 33022-2480
Tel: (954) 925-1100
Fax: (954) 925-1101